

Natural under IMS/TM - Service Modules

This document describes the service modules of the Natural IMS interface. The following topics are covered:

- Purpose of Service Modules
- Service Module Descriptions

Additional Information

- **Natural IMS/TM Error Codes** - for a list of the error codes and messages that may be issued by the Natural IMS Interface (NII) refer to Natural IMS/TM Error Codes in the Natural Messages and Codes documentation.
 - **Installation** - refer to Installing the Natural IMS Interface in the Natural Installation Guide for Mainframes.
 - **Further information** - refer to the following topics:
Environments | Components | Configuration Macros | Service Programs | Special Functions | User Exits | Recovery Handling
-

Purpose of Service Modules

Service modules perform IMS/TM-specific functions. They can be called from within a Natural program using the standard Natural CALL interface. Sample programs are loaded by a Natural INPL into the library SYSEXTP.

Service Module Descriptions

This section contains a detailed description of all the service modules in alphabetical order. This includes a list of the parameters available and the name of the module-relevant sample program.

CMCMMND - Issue IMS Operator Commands

The module CMCMMND issues IMS operator commands and returns the reply segments to the Natural user program.

The following parameters are provided:

Command | Command length | Reply | Length of reply area

Name	Format	Type	Comment
Command		Input	
Command length	(B4)	Input	
Reply		Output	
Length of reply area	(B4)	Input	

The operator command contained in the command area is issued to IMS with the indicated length.

If the user has set a non-zero reply length, any reply segments from IMS are moved into the reply area over the maximum available length. If the reply area is at least two bytes long, the first two bytes contain the IMS status code after the command call has been issued. The two rightmost bytes of the REPLGTH field contain the effective length of the total reply moved into the REPLY field.

If the reply from IMS has to be truncated, this is indicated by setting X'80' in the leftmost byte of the REPLGTH field.

Sample Program: NIPSCMND

CMDEFSW - Deferred Transaction Switch to Natural Transaction Code

The module CMDEFSW performs a deferred transaction switch to a Natural transaction code.

The following parameter is provided:

Trancode

Name	Format	Type	Comment
Trancode		Input	

With the next terminal I/O, the output is sent to the terminal and the next input from this terminal is processed by the transaction code passed as parameter message.

CMDEFSWX - Deferred Transaction Switch to Non-Natural Transaction Code

The module CMDEFSWX performs a deferred switch to a non-Natural transaction code.

The following parameters are provided:

Trancode | Message | Message length | MOD name

Name	Format	Type	Comment
Trancode		Input	
Message		Input	
Message length		Input	
MOD name		Input	

With the next terminal I/O, the given message with the given MOD name is inserted and the Natural session is terminated.

If the new transaction code is a Natural transaction code, the message and the MOD name passed as parameters are ignored and CMDEFSWX works as CMDEFSW.

Sample Programs: NIPSDEFX.

CMDIRNMX - Switch to Another Conversational Transaction w/o Message

The module CMDIRNMX has the same functionality as CMDIRSWX, except that no message is inserted to the alternate PCB. Thus, the only parameter you have to provide is TRANCODE.

The following parameter is provided:

Trancode

Name	Format	Type	Comment
Trancode		Input	

CMDIRNMX can also be used to perform a direct switch to another Natural transaction code, because in this case, the CLEAR key is given as input message to Natural by default.

CMDIRNMZ - Switch to Another Conversational Transaction w. Message

The module CMDIRNMZ has the same functionality as CMDIRSWZ, except that no message is inserted to the alternate PCB. Thus, the only parameter you have to provide is TRANCODE.

The following parameter is provided:

Trancode

Name	Format	Type	Comment
Trancode		Input	

CMDIRSWX - Switch to Another Conversational Transaction w. Message

The module CMDIRSWX performs a direct switch to another conversational transaction and specifies a message that is to be passed on to this new transaction.

The following parameters are provided:

Trancode | Message | Message length

Name	Format	Type	Comment
Trancode		Input	
Message		Input	
Message length	(B4)	Input	

At the next terminal I/O, a change call is executed against the alternate PCB to set its destination to the value of the TRANCODE field. The SPA and the message are then inserted into the alternate PCB.

The new transaction code is checked if it is a Natural or a non-Natural transaction code.

In the case of a non-Natural transaction code, the Natural session is terminated.

In the case of a Natural transaction code, the CLEAR key is passed to Natural as input message, which means that Natural reacts as if the terminal user pressed the CLEAR key. The type of the new transaction code is automatically honored.

Sample Program: NIPSDIRX

CMDIRSWZ - Switch to Another Conversational Transaction w. Message

The module CMDIRSWZ has the same functionality as CMDIRSWX.

The following parameters are provided:

Trancode | Message | Message length

Name	Format	Type	Comment
Trancode		Input	
Message		Input	
Message length	(B4)	Input	

The difference compared to CMDIRSWX is that, in case of a switch to a non-Natural transaction code, the current Natural session is not terminated. This is done with the following intention:

- A given Natural session gives control to a non-Natural transaction code; the session is not terminated.
- The non-Natural transaction performs a terminal I/O and then switches back to the original Natural transaction, passing data into the SPA.
- The Natural transaction does not start a new session, but continues the old session where it has left it, which means that the roll slot is obtained from the swap pool and control is given to Natural so as to continue with an existing session.

The non-Natural transaction code must pass the message "LLZZD", where "LL=H'0005'", "ZZ=X'0000'" and "D=X'6D'" are simulating to Natural that CLEAR has been pressed. By making the Natural program sensitive to the CLEAR key, it is able to recognize that the called non-Natural transaction has come back and it can retrieve the data prepared by the non-Natural transaction for use in subsequent processing.

CMDIRSWZ cannot be used if the transaction code to switch to is a Natural transaction code.

Sample Program: NIPSDIFS

CMDISPCB - Get PCB Content

The module CMDISPCB is used to obtain the contents of a PCB.

The following parameters are provided:

PCB number | Receiving area | Area length

Name	Format	Type	Comment
PCB number	(B4)	Input	
Receiving area		Output	
Area length	(B4)	Input	

After the call is executed, the receiving area contains the contents of the PCB with the requested number in the requested length. A check is made to verify that the requested PCB is within your current PCB list. The first PCB is PCB number 1, the second PCB is PCB number 2, etc.. If you specify an invalid number, the field PCBNUM is set to X'FFFFFFFF' and no further information is passed to your application program.

Sample Program: NIPSPCBD

CMEMOD - Modify MOD Name Dynamically

The module CMEMOD allows the MOD name to be modified dynamically for a given LTERM at the normal end of a Natural session.

The following parameter is provided:

MOD name

Name	Format	Type	Comment
MOD name	(A8)	Input	

At a normal end of a session, the environment-dependent interface inserts the message X'00060000403F' into the IOPCB, using the MOD name whose value is contained in MOD name parameter. This is intended to present a meaningful screen (for example, a general menu) to the terminal user so that he can continue working at the terminal.

CMGETMSG - Read Next Message

The module CMGETMSG reads the next message from the message queue.

The following parameters are provided:

Message area | Message area length

Name	Format	Type	Comment
Message area		Output	
Message area length	(B4)	Input	

The length is checked to see if the received message fits into the message area. The message is moved including the LLZZ bytes into the message area. If there are no more messages, LL=0 is moved into the message area.

If the message does not fit into the message area, a corresponding error message is returned.

Sample Programs: NIPSGETM and NIPSOBMP.

CMGETSEG - Read Next Segment

The module CMGETSEG reads the next segment of the current message from the message queue.

The following parameters are provided:

Message area | Message area length

Name	Format	Type	Comment
Message area		Input	
Message area length	(B4)	Input	

The length is checked to see if the received message fits into the message area. The message segment is moved including the LLZZ bytes into the message area. If there are no more message segments, LL=0 is moved into the message area.

If the message does not fit into the message area, a corresponding error message is returned.

Sample Program: NIPSOBMP

CMGETSPA - Transfer Data from SPA

The module CMGETSPA transfers the data from the SPA starting from the given offset in the requested length into the receiving area.

The following parameters are provided:

Offset | Length | Area

Name	Format	Type	Comment
Offset	(B4)	Input	
Length	(B4)	Input	
Area	(B4)	Output	

Sample Programs: NIPSGSPA and NIPSPSPA

CMIMSID - Get MVS Subsystem ID

The module CMIMSID enables Natural programs to obtain the MVS subsystem ID of the IMS system in which they are currently scheduled.

The following parameter is provided:

IMSID

Name	Format	Type	Comment
IMSID	(A4)	Output	

After the call is executed, the field IMSID contains the MVS subsystem ID of the IMS system in which you are currently scheduled.

The module CMIMSID depends upon an internal IMS control block. Therefore, it is an IMS release-dependent function that will be updated whenever possible.

CMIMSINF - System Environment Info

The module CMIMSINF provides system environment information.

The following parameters are provided:

IMSID | SUFFIX | APPLGNAM | APPLNAM | NRENT | NNONR

Name	Format	Type	Comment
IMSID	(A4)	Output	The IMS ID.
SUFFIX	(A2)	Output	The preload suffix.
APPLGNAM	(A8)	Output	The application group name.
APPLNAM	(A8)	Output	The application name.
NRENT	(B4)	Output	The number of reentrant modules preloaded.
NNONR	(B4)	Output	The number of non-reentrant modules preloaded.

CMIMSINF is also an IMS release-dependent module.

Sample Program: NIPSINF

CMPCBADR - Return PCB Address

The module CMPCBADR returns the address of a PCB which is identified by a logical name. The PSB name is also returned to the Natural program.

The following parameters are provided:

PSB name | PCB name | PCB address

Name	Format	Type	Comment
PSB name	(A8)	Input	
PCB name	(A8)	Input	
PCB address	(B4)	Input	

After the call is executed, the field PCBADR contains the address of the PCB identified in the table module by the logical name "PCBNAME" in the table entry that corresponds to the currently scheduled transaction code. If the logical name does not exist for this transaction code, X'FFFFFFFF' is returned in the PCBADR field. In any case, the field PSBNAME contains the name of the currently scheduled PSB.

Sample Program: NIPSPCBA

CMPRNTR - Change Default Hardcopy Destination

The module CMPRNTR changes the default hardcopy destination set by the module NIIIMSHC to the value passed as parameter.

The following parameter is provided:

Destination

Name	Format	Type	Comment
Destination	(A8)	Input	

The module CMPRNTR is provided for compatibility reasons only; use the Natural SET CONTROL *hdest-id* statement instead.

CMPUTMSG - Insert Output Message into IO-PCB

The module CMPUTMSG can be used to insert any given output message of a given length into the IO-PCB, using any given MFS MOD name.

CMPUTMSG takes the number of bytes as indicated in the message length from the message area and inserts them with the specified MOD name in the message queue.

There is no restriction upon the length of the message, except that it has to fit into the input message area of the environment-dependent interface. No check is made regarding the IO-PCB status code after the insert call is issued to IMS/TM. In this way, you can send MFS-formatted output messages back to the originator of the input message.

The following parameters are provided:

Message area | Message length | MOD name

Name	Format	Type	Comment
Message area		Input	
Message length	(B4)	Input	
MOD name		Input	

CMPUTSPA - Move Data into SPA

The module CMPUTSPA moves the data with the given length at the specified offset into the SPA.

The following parameters are provided:

Offset | Length | Data

Name	Format	Type	Comment
Offset	(B4)	Input	
Length	(B4)	Input	
Data		Input	

A check is done if the specified offset points into the Natural Reserved Area (NRA) within the SPA. If yes, return code 4 is returned.

Sample Program: NIPSPSPA

CMQTRAN - Content of Current Transaction Code Table Entry

The module CMQTRAN returns the contents of the current entry within the transaction code table.

The following parameters are provided:

Transaction code | Offset | Length | Uoffset | PSB name | Number of PCBs

Name	Format	Type	Comment
Transaction code		Output	The transaction code under which you are running.
Offset	(B2)	Output	The offset of the NRA with the SPA.
Length	(B2)	Output	The length of the NRA.
Uoffset	(B2)	Output	Not used.
PSB name		Output	The name of the scheduled PSB.
Number of PCBs		Output	The number of PCBs whose addresses you can obtain using the module CMPCBADR.

The logical names by which you can refer to PCBs in the module CMPCBADR are not returned because of security considerations; you should be informed by your system about which logical names you are allowed to refer to.

Sample Program: NIPSQTRA

CMQUEUE - Insert Message into First Alternate PCB

The module CMQUEUE inserts a message into the first alternate PCB.

The following parameters are provided:

Destination | Message | Message length

Name	Format	Type	Comment
Destination		Input	
Message		Input	
Message length	(B4)	Input	

This call causes an immediate change call to set the destination of the first alternate PCB to the value contained in the field Destination, after which the message is inserted into the alternate PCB with the indicated Message length.

The transaction code is inserted after the LLZZ bytes with a length of 8.

After a PURGE call has been issued, control is returned to the next instruction in the Natural program.

The message can have any length up to the size of the input message area (usually 8000 minus 12 bytes).

Sample Program: NIPSQLOA

CMQUEUEX - Complete Control over Message Content

The module CMQUEUEX provides you with complete control over the contents of a message that is to be queued in the IMS/TM input queue.

The following parameters are provided:

Destination | Message | Message length

Name	Format	Type	Comment
Destination		Input	
Message		Input	
Message length	(B4)	Input	

This call causes an immediate change call to set the destination of the first alternate PCB to the value contained in the field Destination, after which the message is inserted into the alternate PCB with the indicated Message length after the LLZZ bytes. The difference compared to CMQUEUE is that the transaction code is **not** inserted after the LLZZ bytes.

After a PURGE call has been issued, control is returned to the next instruction in the Natural program. The message can have any length up to the size of the input message area (usually 8000 minus 12 bytes).

Sample Program: NIPSQUEX

CMSNFPRT - Set Logical Device Name

The module CMSNFPRT sets the logical name of the device to which the Natural messages during the online BMP run is sent.

The following parameter is provided:

Printer name

Name	Format	Type	Comment
Printer name		Input	

Before calling CMSNFPRT, use the Natural parameter SENDER to define the default output destination.

Sample Program: NIPSOBMP

CMSVC13D - Terminate Natural Session

The module CMSVC13D terminates the Natural session with user abend 222 and creates a dump.

Parameters: None

Sample Program: None.

CMTRNSET - Insert SPA via Alternate PCB

When the Natural session is terminated normally, the Natural IMS interface performs a direct program-to-program switch to the specified transaction code and inserts the SPA via the alternate PCB.

The following parameter is provided:

Trancode

Name	Format	Type	Comment
Trancode		Input	

Sample Program: NIPSEOSS

NIIDDEFS - Deferred Switch to Foreign Transaction

Module NIIDDEFS is similar to module CMDEFSWX. If you use NIIDDEFS to perform a deferred switch to a foreign transaction, the current Natural session is suspended, as with module CMDIRSWZ. The suspended Natural session can be resumed at any time by sending back to Natural a message containing the CLEAR key.

The following parameters are provided:

Transaction code | Message | Message length | MOD name | Transaction type

Name	Format	Type	Comment
Transaction code		Input	The transaction code to switch to.
Message		Input	The message to be sent to the foreign transaction code.
Message length	(B4)	Input	
MOD name	(A8)	Input	
Transaction type	(A4)	Input	An A4 variable containing the string "CONV" if the foreign transaction is conversational and the string "NONC" if the foreign transaction is non-conversational.

Return Codes:

- 0 OK
- 4 The message length is greater than the size of the message area defined in the environment table.
- 8 You tried to do a deferred switch with suspend from a conversational Natural to a non-conversational foreign transaction, something which cannot be done.
- 12 The fifth parameter is invalid; it contains neither "CONV" nor "NONC".

Sample Program: NIPSDEFS

NIIDPURG - Insert Multi-Segment Message

The module NIIDPURG does not have parameters. It issues a PURG call using the first alternate PCB and inserts multi-segment messages using the module NIIDQUMS.

Return Codes: Either bytes two and three of the 4-byte return code contain the status code, or the return code has the value 0.

Sample Program: NIPSQLMS

NIIDQUMS - Create Multi-Segment Message

This module creates multi-segment messages. It has basically the same functionality as the module CMQUEUE, with the difference that NIIDQUMS does not issue a PURG call.

The following parameters are provided:

Destination | Message | Message length

Name	Format	Type	Comment
Destination		Input	
Message		Input	
Message length	(B4)	Input	

It is your responsibility to issue the PURG call using the module NIIDPURG.

Sample Program: NIPSQLMS

NIIDSETT - Get Foreign Transaction Code

In order to perform a correct transaction switch to a foreign transaction code, the type of the foreign transaction code must be known. To obtain this type, the special-purpose module NIIDSETT can be used. If NIIDSETT is not used, the foreign transaction code is assumed to be of the same type as the invoking Natural transaction code. If this is not the case, there will be unpredictable results or the session will terminate abnormally.

The following parameter is provided:

Transaction type

Name	Format	Type	Comment
Transaction type	(A4)	Input	Possible values: "CONV" for conversational, "NONC" for non-conversational.